REMARKS/ARGUMENTS

Claims 1-11 are pending in the application. Claims 1-11 have been rejected.

Rejections under 35 U.S.C. §102.

The Office Action has rejected claims 1-11 under 35 U.S.C. §102(e) as being anticipated by a publication by U.S. Patent No. 6,360,360 tilted "Object-Oriented Compiler Mechanism for Automatically Selecting Among Multiple Implementation of Objects" (hereafter, "Bates"). Applicant respectfully traverses the rejection for the following reasons.

Claim 1 relates to a method of performing a) providing said component with a plurality of explicit selectable alternative implementations which share a common component interface and semantics; b) instrumenting said component to gather cost-related information during at least a partial run of said program; c) providing said component with a cost estimator for using said cost-related information to estimate a cost for using each of said explicitly selectable implementations in running said program; d) based on the costs estimated in step (c), selecting one of said explicitly selectable implementations for a subsequent at least partial run of said program.

Bates relates to an object-oriented method and apparatus for automatically selecting among multiple implementations of objects in a computer program. In an object-oriented or object-based computer system, Bates discloses a compiler mechanism that allows a compiler to automatically select among multiple implementations of an object to optimize the performance of the compiled code. The selection may be made by assigning a weighted cost to each of the implementations, and selecting the implementation with the lowest weighted cost, where the weighted cost represents estimates of execution frequencies for each of the object's functions. In the alternative, for implementations that have different interfaces, the selection may be made based on an evaluation of the functions used in the program compared to the functions provided by the different implementations, and selecting an implementation that contains all functions issued against the object with a minimum of functions that are not issued against the object.

By not showing how Bates teaches all claim limitations, the Office Action fails to make a prima facie case of anticipation.

In the Office Action of November 17, 2004, the Examiner stated that the Bates reference discloses "instrumenting said component to gather cost-related information during at least a partial run of said program" and points the Applicant to col. 8 lines 33-35 which purportedly describes such a step. A read of the Bates passage does not describe such a step. The cited Bates passage reads:

> "The test code is compiled (step 810), and run on a target platform (step 820) (i.e., a computer that has the same configuration as the computer of the anticipated user of the code). During execution of the test code, profile data is gathered as the code runs to track the number of times the test code actually executes each function in each object. This profile data may be acquired by any suitable technique, such as inserting instrumentation code or acquiring an address trace as the test code executes."

Thus, the cited Bates passage makes no mention of instrumenting of a program component to gather cost-related information during at least a partial run of the program, as defined in the independent claims of Applicant's invention. The Applicant describes in detail the cost functionality of the Applicant's invention in the specification, namely on page 15 lines 10-20 (among other locations in the specification):

> "The implementor is responsible for providing cost functions that return cost of each call in the different implementations, so the cost of the successor query for TreeSet should return log of the size of the tree. If two variables contain sets, there is a cost of cloning from one to the other, namely the cost of the copy. In our model the user who invokes the clone operation should not know the underlying representation. So, as part of a clone operation the system will invoke the correct transfer function, say tree2hashset in addition to making the copy, There is a cost of doing that additional conversion and in addition to writing the tree2hashset function the author of the multiple implementations must provide the coercion cost function so the system can understand the penalty of using different implementations."

Thus, the Bates reference does not disclose the instrumenting of a program component to gather cost-related information during at least a partial run of the program, as claimed in Applicant's invention. For this reason, the Bates reference does not disclose, teach, or suggest the aforementioned elements of independent claims 1, 5 and 9 - namely, the instrumenting of a program component to gather cost-related information during at least a partial run of the program. Thus, the Examiner's rejection of these claims has been traversed and the Applicant respectfully requests that the rejection is withdrawn. The Applicant further requests allowance of these claims.

Further, dependant claims 2-4, 6-8 and 10-11 depend from and include all of the limitations of independent claims 1, 5 and 9. For this reason, the Applicant respectfully requests withdrawal of the Examiner's rejection and allowance of these claims.

Rejections under 35 U.S.C. §103

The Office Action rejected claims 1-11 as unpatentable over U.S. Patent 6,324,619 (Raverdy et al) in view of Blake et al (U. S. Patent 5,752,038, hereafter Blake). The Applicant respectfully disagrees.

The Office Action admits that Raverdy neither teaches nor suggests the claim limitation of instrumenting of a program component to gather cost-related information during at least a partial run of the program. Further, the Office Action does not show the presence of this step in the Blake reference.

Blake relates to a method and system for determining an optimal placement order for code portions within a module to improve locality of reference and reduce the working set of the module. Blake discloses the reduction of the working set of a module. The optimal placement order for code portions within a module reflects the concurrency of usage for code portions during execution of the module. That is, all code portions which execute within a certain period of time are placed in close proximity to each other within the executable module. This method of "time ordering" reduces the working set of a module

In the Office Action of November 17, 2004, the Examiner stated that the Blake reference discloses "instrumenting said component to gather cost-related information during at least a partial run of said program" and points the Applicant to col. 2 lines 45-47 which purportedly describes such a step. A read of the Blake passage does not describe such a step. The cited Blake passage reads:

> "When determining the optimal placement order for each code portion, the present invention executes an instrumented version of the module to collect execution data for each code portion, analyzes the execution data to determine the optimal placement order for each code portion, and links the code portions

> according to the determined optimal placement order. The instrumented version of the module contains instructions that, when executed, cause execution data to be recorded. When the code portions are linked according to the determined optimal placement order, the working set for the module is reduced, thereby lessening page and cache misses and improving overall system performance."

Thus, the cited Blake passage makes no mention of instrumenting of a program component to gather cost-related information during at least a partial run of the program, as defined in the independent claims of Applicant's invention. The Applicant describes in detail the cost functionality of the Applicant's invention in the specification, namely on page 15 lines 10-20 (see excerpt above).

Thus, the Blake reference also does not disclose the instrumenting of a program component to gather cost-related information during at least a partial run of the program, as claimed in Applicant's invention. For this reason, neither the Blake reference, the Raverdy reference or any combination of the two disclose, teach, or suggest the aforementioned element of independent claims 1, 5 and 9 - namely, the instrumenting of a program component to gather cost-related information during at least a partial run of the program. Thus, the Examiner's rejection of these claims has been traversed and the Applicant respectfully requests that the rejection is withdrawn. The Applicant further requests allowance of these claims.

Further, dependant claims 2-4, 6-8 and 10-11 depend from and include all of the limitations of independent claims 1, 5 and 9. For this reason, the Applicant respectfully requests withdrawal of the Examiner's rejection and allowance of these claims.

For the foregoing reasons, Applicant respectfully requests allowance of the pending claims and that a timely Notice of Allowance be issued in this case.

Respectfully submitted,

Michael J. Buchenhorner

Michael J. Buchenhorner

Reg. No. 33,162

Date: February 17, 2005

HOLLAND & KNIGHT LLP

Holland & Knight LLP
701 Brickell Avenue, Suite 3000
Miami, FL 33131
(305) 789-7773 (voice)
(305) 789-7799 (fax)

Certificate of Facsimile Transmission

I hereby certify that this Amendment and Response to Office Action, and any documents referred to as attached therein are being mailed via Express Mail on this date, February 17, 2003, to the Commissioner for Patents, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450.

Michael J. Buchenhorner

Michael J. Buchenhorner

Date: February 17, 2005